Scalable and Effective Deep CCA via Soft Decorrelation

Scalable and Effective Deep CCA via Soft Decorrelation

Xiaobin Chang[1], Tao Xiang[1], Timothy M. Hospedales[2]
Queen Mary University of London[1], The University of Edinburgh[2]
{x.chang, t.xiang}@qmul.ac.uk t.hospedales@ed.ac.uk

Abstract

Recently the widely used multi-view learning model, Canonical Correlation Analysis (CCA) has been generalised to the non-linear setting via deep neural networks. Existing deep CCA models typically first decorrelate the feature dimensions of each view before the different views are maximally correlated in a common latent space. This feature decorrelation is achieved by enforcing an exact decorrelation constraint; these models are thus computationally expensive due to the matrix inversion or SVD operations required for exact decorrelation at each training iteration. Furthermore, the decorrelation step is often separated from the gradient descent based optimisation, resulting in sub-optimal solutions. We propose a novel deep CCA model Soft CCA *to overcome these problems. Specifically, exact decorrelation is replaced by soft decorrelation via a mini-batch based Stochastic Decorrelation Loss (SDL) to be optimised jointly with the other training objectives. Extensive experiments show that the proposed soft CCA is more effective and efficient than existing deep CCA models. In addition, our SDL loss can be applied to other deep models beyond multi-view learning, and obtains superior performance compared to existing decorrelation losses.*

1. Introduction

Canonical Correlation Analysis (CCA) [12, 7] is widely used for multi-view learning. These views could be camera views, e.g., the images of a face from different view angles, or modalities, e.g., an image and its caption. CCA aims to learn a joint embedding space where different views of a single data item are maximally correlated/aligned. Many tasks can be accomplished in this space such as cross-view recognition, retrieval and synthesis [8, 49, 14, 1, 36, 2].

A standard CCA model is linear in the sense that the projection between the feature space and the embedding space is linear. For learning richer non-linear embeddings, Kernel CCA (KCCA) [10] extended linear CCA via kernelisation. Both linear CCA and KCCA are shallow models and the training procedure usually requires accessing the whole batch data. As a result, KCCA has poor scalability. The recently proposed deep CCA [1, 36, 41, 37] aims to learn nonlinear projections with deep neural networks rather than kernels and has been shown to be more effective than shallow CCA and KCCA.

However, scalability issues remain for deep CCA. This is because existing deep CCA models [1, 36, 41, 37] aim to implement an exact or 'hard' decorrelation. More precisely, before being projected into the common embedding space, the extracted deep feature vector for each view is decorrelated by forcing its correlation matrix over the training batch to be an identity matrix. This decorrelation operation is exact but computationally expensive. Either matrix inversion [1, 36] or singular value decomposition (SVD) [37] is required *at each iteration* which severely limits scalability. Furthermore, existing deep CCA models such as [37] typically employ two separate and independent optimisation steps: The feature representation for each data view is first decorrelated exactly as described above. These decorrelation operations do not directly affect the following gradient computation and subsequent backpropagation. Without jointly optimising the decorrelation constraint and other learning objectives, this could lead to sub-optimal solutions.

In this paper, we propose Soft CCA, a novel approach to deep CCA. In our model, decorrelation is formulated as a soft constraint to be jointly optimised with other training objectives. Specifically, a robust decorrelation loss, called Stochastic Decorrelation Loss (SDL), is introduced, which is mini-batch based and approximates the full-batch statistics efficiently and effectively by using stochastic incremental learning. SDL is a softer constraint as the loss is only minimised rather than enforced to be zero. Comparing with existing deep CCA models, Soft CCA has two advantages: First, it is more efficient and scalable – by avoiding computationally expensive operations such as SVD, its cost is quadratic $O(k^2)$ rather than cubic $O(k^3)$ with a k-dimensional feature input. Second, by jointly optimising the decorrelation loss with other losses such as the distance between views in the embedding space, more globally optimal solutions can be achieved resulting in more effective

correlation analysis and learning of multi-view embeddings.

While our proposed SDL is motivated by the feature decorrelation required for deep CCA learning, it can also be applied as an activation regularisation to any deep model where feature decorrelation is helpful. In this work, we demonstrate this with two widely used models including Factorisation Autoencoder (FAE) and convolutional neural network (CNN) based classifiers. FAE architectures aim to disentangle latent factors of variation that correspond to different aspects of data items. Here we use SDL-based decorrelation to ensure representations of distinct factors are indeed disentangled, and show that it provides superior disentangling performance compared to prior approaches. As for the supervised CNN classifier, it was recently shown that decorrelation losses can be beneficial for maximizing model capacity and reducing overfitting [5]. In this case, we show that by whitening the computed deep features in supervised CNN classifiers, we can train a more effective classifier for both instance and category-level recognition benchmarks.

We conduct extensive experiments on multi-view correlations analysis. The results show that the proposed soft deep CCA is much more efficient as well as more effective than the existing shallow or deep CCA models – and is also simpler to implement. Moreover, we demonstrate that SDL can be applied to a number of models for problems beyond multi-view learning, and improves model performance beyond that of existing decorrelation losses.

2. Related Work

2.1. Deep CCA

Canonical Correlation Analysis (CCA) [12] and its variants including Kernel CCA [10] and multi-view CCA [8] are one of the most popular multi-view learning approaches. Inspired by the success of Deep Neural Network (DNNs) in representation learning [48], Deep CCA has received increasing interest [1, 36, 37]. A deep CCA architecture was first proposed by Deep CCA (DCCA) [1] which directly computes the gradients of CCA objective and requires both a second-order optimisation method [25] and full-batch training inputs. It thus cannot cope with large training data sizes. An alternative deep CCA objective and architecture are proposed in Stochastic Deep CCA (SDCCA) [37] which make it suitable for mini-batch stochastic optimisation. However, due to the exact decorrelation used, SDCCA still requires a costly SVD operation at each training iteration. SVD's $O(k^3)$ cost is not scalable to the large layer sizes k (e.g., $k = 1024$) common in contemporary DNNs. In fact, all existing deep CCA models [1, 36, 37] take an exact decorrelation step, which limits their scalability and effectiveness as mentioned earlier. Furthermore, the exact decorrelating operations often do not directly impact the following gradient computations and backpropagation, which could lead to sub-optimal optimisation. In contrast, our Soft Deep CCA decorrelates by formulating the decorrelation constraint as a loss which is optimised end-to-end jointly with other losses in a standard SGD procedure, making it both more scalable and more effective.

2.2. Decorrelation Loss

Beyond multi-view learning, many other deep models benefit from decorrelation of activations in a neural network layer. For these models, a decorrelation loss such as the proposed SDL can be employed. Two such models are studied in this work, namely the Factorisation Autoencoder (FAE), and convolutional neural network (CNN) based classifiers. For each model, an alternative decorrelation loss exists.

FAE and XCov loss Recently interest has regrown in models for disentangling the underlying factors of variation in the appearance of objects in images, for example identity and viewpoint [49, 38, 15, 34, 16, 24, 23]. FAE achieves semi-supervised disentangling of latent factors via a two-branch autoencoder. Recently it has been shown in [4] that the efficacy of FAE can be improved by adding a decorrelation loss (termed XCov in [4]) to explicitly decorrelate the computed latent factor representations. Like our SDL, computing XCov is also a mini-batch operation. But it only eliminates correlations across and not within each factor; and it computes covariance only within each mini-batch, while our SDL approximates full-batch statistics using stochastic incremental learning. We show in our experiments (Sec. 4.1) that SDL is more effective than XCov for helping FAE to disentangle latent factors.

CNN Classifier and DeCov loss Using CNN with a classification loss (e.g., cross entropy) for object recognition is perhaps the most popular application of deep learning in computer vision. CNN classifiers are used for not only object category recognition tasks [17, 18] but also object instance/identity recognition/verification tasks such as face verification [28] and person re-identification [39]. When training CNNs for classification, avoiding overfitting, saturation and slow convergence are crucial [6]. These problems are often alleviated by regularisation such as Batch Normalisation [13] and dropout [27]. Recently it was shown that decorrelation losses can also be used for effective overfitting reduction [5]. Compared with the existing decorrelation loss DeCov [5], our SDL has the following advantages: (1) More accurate covariance statistics due to full-batch approximation instead of the pure mini-batch statistics used in DeCov [5]. (2) SDL uses a more robust L_1 formuation instead of the L_2 one in DeCov [5], which encourages sparser correlation and thus stronger decorrelation.

Our contributions are as follows: (1) We provide a new perspective on CCA that allows its objective to be expressed as a loss to be minimised by gradient descent rather than as an eigen-decomposition problem. (2) We propose Soft

CCA, a novel Deep CCA model that is simple to implement, more efficient and scalable (mini-batch SGD-based optimisation) and more effective (full batch approximation, jointly end-to-end) than existing deep CCA models. (3) Beyond multi-view learning, our SDL is applicable to a variety of tasks and models, and is superior to alternative decorrelation losses including XCov and DeCov.

3. Soft CCA

3.1. Deep CCA

Deep CCA extends linear CCA model by projecting views of the same item (here we consider images of the same objects) from different views to a common latent space using a DNN with multiple branches, each corresponding to one view (see Fig. 1).

We consider a two-view case for simplicity of notation, but the multi-view extension is straightforward. Assume we have $2N$ images consisting of two views for each of N objects. They are then organised into mini-batches of M image pairs and fed into the two DNN branches. The training images in the two views are denoted as X_1 and X_2 respectively. The DNN branches aim to learn functions that project paired input images into a shared latent space where they are maximally correlated. Denote the DNN projection function for view i, $i = \{1, 2\}$ as $P_{\theta_i} : X_i \rightarrow Z_i$, or $P_{\theta_i}(X_i) = Z_i$ where $Z_i \in \mathbb{R}^{M \times k}$ is the projected feature matrix for M data items for view i in the k-D CCA embedding space and θ_i are the DNN parameters.

Following [7], CCA can be formulated in multiple ways and the most relevant one here is:

$$\begin{aligned} &\arg\max_{\theta_1,\theta_2} Tr(P_{\theta_1}^T(X_1)P_{\theta_2}(X_2)), \\ &s.t.\ P_{\theta_1}^T(X_1)P_{\theta_1}(X_1) = P_{\theta_2}^T(X_2)P_{\theta_2}(X_2) = I, \end{aligned} \tag{1}$$

where I indicates the identity matrix. The constraints enforce decorrelation within each of the two input signals. Eq. 1 can be written into an equivalent form:

$$\begin{aligned} &\arg\min_{\theta_1,\theta_2} \frac{1}{2}||P_{\theta_1}(X_1) - P_{\theta_2}(X_2)||_F^2, \\ &s.t.\ P_{\theta_1}^T(X_1)P_{\theta_1}(X_1) = P_{\theta_2}^T(X_2)P_{\theta_2}(X_2) = I, \end{aligned} \tag{2}$$

where $||\cdot||_F$ is the Frobenius norm of a matrix. It shows that the goal of maximising correlation between $P_{\theta_1}(X_1)$ and $P_{\theta_2}(X_2)$ can be achieved by minimising the L_2 distance between the decorrelated signals.

The key idea of our approach is to convert the hard constraint in Eq. 2 into a soft cost to be optimised by SGD.

3.2. Stochastic Decorrelation Loss (SDL)

We denote the representations from one branch of a deep CCA network over a mini-batch as $Z \in \mathbb{R}^{m \times k}$, where

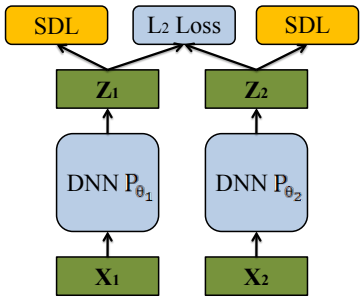


Figure 1: Schematic of implementing Soft CCA with SDL.

m is the mini-batch size and k indicates the number of neurons/feature channels. We further assume that Z has been batch-normalised, i.e., each activation over the mini-batch has zero mean and unit variance. This can be easily achieved by adding a Batch Normalisation (BN) [13] layer.

The mini-batch covariance matrix C_{mini}^t for the t-th training step then is given as:

$$C_{mini}^t = \frac{1}{m-1} Z^T Z. \tag{3}$$

However, full-batch statistics are required by CCA objective for decorrelation. Therefore, we approximate the full-batch covariance matrix C_{full} by accumulating statistics collected from each mini-batch. This is achieved by stochastic incremental learning. More specifically, we first compute an accumulative covariance matrix:

$$C_{accu}^t = \alpha C_{accu}^{t-1} + C_{mini}^t, \tag{4}$$

where $\alpha \in [0, 1)$ is a forgetting/decay rate and C_{accu}^0 is initialised with an all-zero matrix. A normalising factor is also computed accumulatively as $c^t = \alpha c^{t-1} + 1$ ($c^0 = 0$ initially). The final full-batch covariance matrix approximation is then computed as:

$$C_{appx}^t = \frac{C_{accu}^t}{c^t}. \tag{5}$$

If we were to follow an exact decorrelation strategy as in [1, 36, 37], we need to force the off-diagonal elements of C_{appx}^t to zero. However, that has implications on the computational cost and scalability which we shall detail later. Instead, we follow a soft decorrelation procedure and formulate the decorrelation constraint as a loss. Specifically, SDL is an L_1 loss on the off-diagonal element of C_{appx}^t:

$$L_{SDL} = \sum_{i=1}^{k} \sum_{j \neq i}^{k} |\phi_{ij}^t|, \tag{6}$$

where ϕ_{ij}^t is the element in C_{appx}^t at (i, j). L_1 loss is used here to encourage sparsity in the off-diagonal elements. SDL is soft because it only penalises the correlation across

activations instead of enforcing exact decorrelation. It will be jointly optimised with any other losses the model may have.

Gradients and Optimisation The gradient of L_{SDL} w.r.t. z_{ni} (the element in Z at (n,i)) can be computed as

$$\frac{\partial L_{SDL}}{\partial z_{ni}} = \frac{1}{c^t}\frac{1}{m-1}\sum_{j}^{k} S(i,j) z_{nj}, \\ S(i,j) = \begin{cases} 1, \ \phi_{ij}^t > 0 \\ 0, \ i=j \ or \ \phi_{ij}^t = 0 \\ -1, \ \phi_{ij}^t < 0 \end{cases} \tag{7}$$

with the sign matrix $S \in \mathbb{R}^{k \times k}$ and $i, j = 1, ..., k$. Eq. 7 can be written in a matrix form:

$$\frac{\partial L_{SDL}}{\partial Z} = \frac{1}{c^t}\frac{1}{m-1} Z \cdot S, \tag{8}$$

where $\cdot$ indicates matrix multiplication.

Once the SDL gradients are computed, they are passed through the network during back-propagation and optimised along with other losses in end-to-end training.

3.3. Computational Complexity

Eq. 6 shows that to compute the SDL in a forward pass, we need matrix multiplication (as in Eq. 3), matrix addition (as in Eq. 4) and matrix element-wise summation (as in Eq. 6). Therefore, the forward pass computation complexity of SDL is $O(mk^2)$. The gradient computation during the backward pass is in Eq. 8. It is also a matrix multiplication and therefore the complexity is $O(mk^2)$. The overall computational complexity of one training iteration is thus $O(mk^2)$. In contrast, existing exact decorrelation computation [1, 37] has a complexity of $O(mk^2 + k^3)$ due to the use of SVD. Note that in large scale vision problems, the number of activations in an FC layer can easily be thousands, meaning that the alternative hard decorrelation models are prohibitively expensive.

3.4. SDL for Soft CCA

With the proposed SDL, the constrained optimisation problem in Eq. 2 can be reformulated as the following unconstrained objective:

$$\arg\min_{\theta_1,\theta_2} L_{dist}(P_{\theta_1}(X_1), P_{\theta_2}(X_2)) \\ + \lambda(L_{SDL}(P_{\theta_1}(X_1)) + L_{SDL}(P_{\theta_2}(X_2))), \tag{9}$$

where $L_{dist}(P_{\theta_1}(X_1), P_{\theta_2}(X_2))$ is the L_2 distance and λ weights the alignment versus decorrelation losses. The Soft CCA architecture is also illustrated in Fig. 1. Note that both SDL and L_2 loss are mini-batch based losses. Therefore, Soft CCA (deep CCA model with SDL) can be realised using standard SGD optimisation for end-to-end learning.

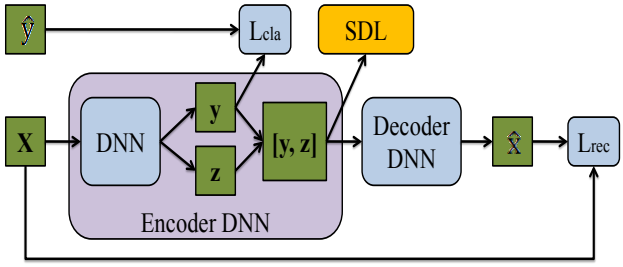


Figure 2: Architecture of FAE with SDL.

4. Applications of SDL to other deep models

4.1. Factorisation Autoencoder with SDL

We describe a two-factor case although the model generalises to an arbitrary number of factors. The two-factor FAE model is illustrated in Fig. 2. Its encoder (a deep neural network) takes image x as input and projects it into an embedding space/latent code which has two parts: y and z. We assume y is a factor that is annotated in the training data, e.g., class label. The other unspecified factors are thus captured by z. Both y and z are used as input to the decoder (e.g., a deconvolutional network) which produces a reconstruction of x, denoted as $\hat{x}$. The goal is not only to accurately reconstruct the input x, but also to represent distinct factors of variation in y and z (e.g., class and style respectively).

Assume the FAE model is parameterised by θ. Given a training set D containing images X and their labels $\hat{Y}$ for the known factor, the learning objective of FAE is:

$$\arg\min_{\theta} L_{rec}(X, \hat{X}) + \lambda L_{cla}(Y, \hat{Y}), \tag{10}$$

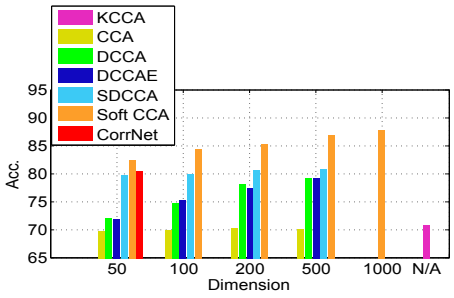
where $L_{rec}(X, \hat{X})$ is the reconstruction loss, which we use pixel L_2 loss here, and $L_{cla}(Y, \hat{Y})$ is the classification loss, i.e., cross-entropy loss here. If there is no constraint on the relation between y and z, they would not necessarily represent distinct aspects of the input signal. To disentangle them, we introduce our SDL to the objective:

$$\arg\min_{\theta} L_{rec}(X, \hat{X}) + \lambda_1 L_{cla}(Y, \hat{Y}) + \lambda_2 L_{SDL}([Y, Z]). \tag{11}$$

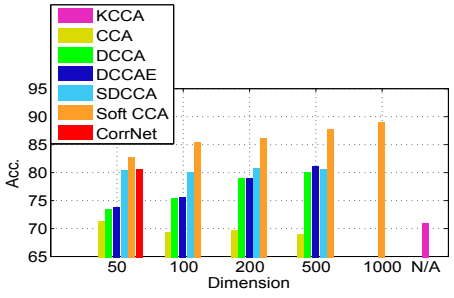
As shown in Fig. 2, this means we decorrelate the elements of the concatenated code $[y, z]$ which decorrelates the two code parts (factors), as well as the signal within the factors.

4.2. CNN Classifier with SDL

Since decorrelation loss encourages a layer's activations to be decorrelated, it reduces activation co-adaptation and maximises the model's capacity. Therefore, SDL can be applied to each layer of a CNN classifier to boost the model performance. In our experiments, we add SDL to different

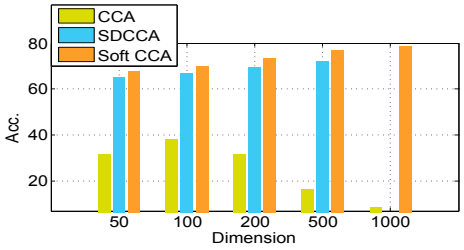


(a) Left-to-right

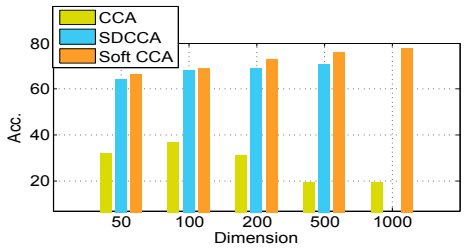


(b) Right-to-left

Figure 3: Cross-view digit recognition results on MNIST. Note that CCA is not scalable to a common space dimension that is greater than the total dimension of 784. Moreover, DCCA, DCCAE and SDCCA are also intractable with our GPU resources when the common space dimension becomes 1000.



(a) Left-to-right



(b) Right-to-left

Figure 4: Cross-view face recognition results on Multi-PIE. Accuracy (%). Note that SDCCA is intractable with our GPU resources when the common space dimension becomes 1000.

CNN classifiers for different recognition tasks to demonstrate its general applicability.

5. Experiments

5.1. Soft CCA

Datasets and settings We evaluate the proposed Soft CCA and alternative deep CCA models on two widely used datasets. **MNIST** [19] consists of handwritten digit images with an image size of 28 $\times$ 28. It contains 60,000 training and 10,000 testing images respectively. We follow the experimental setting in [2] for cross-view recognition. Deep CCA models are trained on the left and right halves of a 10,000 sized subset of training images and we do 5-fold cross validation on the provided test set for recognition. **Multi-PIE** [9] is a face dataset composed of 750,000 images of 337 people with various factors contributing to appearance variation including viewpoint, illumination and facial expression. We use a subset containing 6,200 images of all 337 identities in neutral expression and lighting. Constructing an analogous experiment to the cross-view recognition benchmark, these images are separated into the left and right view groups according to their viewing angle. Left-right view angle pairs are then formed exhaustively for the same identities to train the deep CCA models. We use half of the images in both views for deep CCA training and also do 5-fold cross validation for recognition on the rest of the data.

Implementation details For MNIST cross-view recognition, the network architecture of each view branch is identical to that in [2] for fair comparison. Concretely, there are three hidden layer containing 500, 300, k units/activations respectively, where the k units are used as the common representation (CCA embedding layer). ReLU is applied on the hidden layers' activations (except the embedding layer). Once the CCA model is trained, on the test set, features from one view (e.g., right) are exacted, embedded with deep CCA, and then fed to a Linear SVM [3] classifier which is trained to recognise the images. Finally, the model is evaluated based on features from the other view (e.g., left) being projected into the shared embedding space, and recognised by the SVM. Clearly, the performance of the SVM on this cross-view recognition task depends on the efficacy of the CCA embedding. An analogous cross-view recognition setting is used for the Multi-PIE dataset. The DNN architecture for Multi-PIE also has three hidden layers: 1024, 512,

	50D	100D	200D	500D	1000D
Upper Bound	50	100	200	500	1000
CCA [12]	28.3	34.2	48.7	74.0	-
DCCA [1]	29.5	44.9	59.0	84.7	-
DCCAE [35]	29.3	44.2	58.1	84.4	-
SDCCA [37]	**46.4**	**89.5**	166.1	307.4	-
Soft CCA	45.5	87.0	**166.3**	**356.8**	**437.7**

Table 1: Correlation strength on MNIST. '-' indicates that the result is not obtainable due to the corresponding model being intractible with our available hardware.

k units, the k units are used as the CCA embedding layer. ReLU is applied on the hidden layers' activations (except the embedding layer).

Competitors For shallow CCA, we compare the standard linear CCA [12] and its nonlinear kernelised variant, KCCA [10]. The KCCA results are obtained from [2]. For the deep CCA models, we compare with CorrNet [2], DCCA [1], DCCAE [35] and SDCCA [37]. CorrNet [2] combines correlation maximisation with cross-view autoencoder loss and uses Batch Normalisation. Without access to their code, we can only use the reported result in [2] which was obtained only on MNIST with $k = 50$. As far as we know, SDCCA [37] is the most efficient state-of-the-art deep CCA model to date.

Results on cross-view recognition Figures 3 and 4 show the results for cross-view digit and face recognition. We make the following observations: (1) The deep models achieve better performance than the shallow ones. (2) Our Soft CCA achieves the best results on both datasets with all CCA space dimensions. (3) Increasing the common space dimension k benefits SDCCA very little and even harms the performance of other competitors (e.g. CCA). In contrast, our Soft CCA clearly benefits from larger CCA space dimensions.

Results on cross-view correlation Another way to evaluate CCA models is to measure the average correlation strength of each matching pair of data when they are projected into the common CCA space [37]. We follow the experimental setting and network architecture of [37] (SDCCA) for a fair comparison. The results of MNIST and Multi-PIE are shown in Table 1 and Table 2 respectively. We can conclude from the results that: (1) Again the deep models achieve higher correlation values indicating that they align the two views much better than the linear CCA model. (2) For the easier digit classification task in MNIST, our model is slightly inferior to SDCCA at 50D and 100D, but better after 200D. For the more challenging face recognition problem in Multi-PIE, Soft CCA consistently outperforms SDCCA and the gap increases with the dimension.

	50D	100D	200D	500D	1000D
Upper Bound	50	100	200	500	1000
CCA [12]	12.8	23.9	53.4	140.6	207.1
SDCCA [37]	25.7	51.5	151.2	228.3	-
Soft CCA	**29.2**	**60.5**	**163.2**	**257.7**	**283.9**

Table 2: Correlation strength on Multi-PIE.

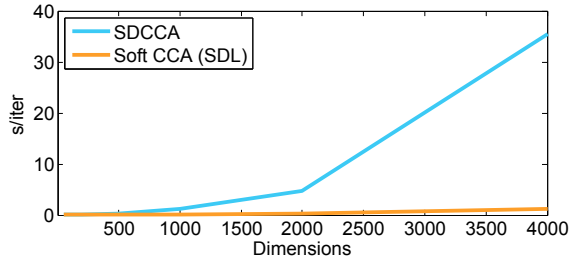


Figure 5: Comparing training time (seconds/iteration) on MNIST given different CCA space dimensions.

These results suggest that our model is more effective with higher dimensional embedding space, which is required for more challenging computer vision tasks.

Evaluation on scalability We compare the training time for our model and that for the most efficient deep CCA model proposed to date, SDCCA [37]. Figure 5 shows that our soft CCA is always more efficient than SDCCA even at the low dimensions[1]. Importantly, when the CCA embedding space dimension approaches 4,000 (roughly the same as the final FC layer size of popular DNNs like AlexNet and VGGNet), our model is clearly much more efficient to train. This is due to the $O(k^2)$ vs. $O(k^3)$ computational complexity difference.

5.2. FAE with SDL

Dataset and settings We use MNIST [19], and follow the same experimental setting as [4]. The network architecture is 784-1000-1000-{y+z}-1000-1000-784, where 784 is the dimension of the vectorised image. ReLU is applied on the hidden layers' activations (except y, z). As shown in Fig. 2, among the two factors to be disentangled, y is the digit class which is annotated with the training data. The other factor z corresponds to aspects of appearance besides class – i.e., the unannotated writing style. In our experiments, the dimension of y is fixed to 10 corresponding to the 10 digit classes and the dimension of z is also set to 10. We compare the performance of a vanilla FAE (basic network with only reconstruction and classification loss),

[1]The speedup is significant even under low dimensions; it is just not very salient in Fig. 5 due to the scaling problem. E.g, at 50D and 100D, Soft CCA is 2 and 5 time faster to train respectively.

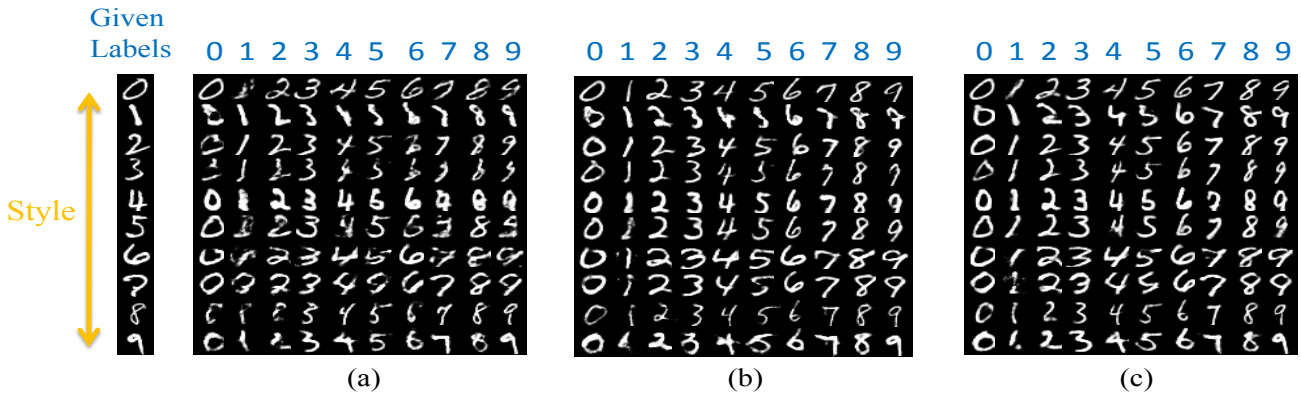


Figure 6: Qualitative results of handwriting style transfer with different FAE models. (a) FAE; (b) FAE + XCov [4] ; (c) FAE + SDL. The dimension of z is set to 10.

	FAE	XCov [4]	DeCov [5]	SDL
z ($\downarrow$)	43.44	14.51	15.42	**11.35**
y ($\uparrow$)	97.23	95.72	97.09	**97.33**

Table 3: Disentanglement efficacy. Classification accuracy (%) using representation of each branch in MNIST FAE.

	Accuracy
Baseline [11]	91.12
DeCov [5]	91.62
SDL	**92.44**

Table 4: CIFAR10 classification results (%)

FAE+XCov [4], FAE+DeCov [5] and our FAE+SDL.

Evaluation on disentanglement In the ideal case, the two factors will be completed disentangled in y and z, i.e., y contains no information about the style and z contains nothing about the class. To quantify this, we compare the digit classification performance with the inferred y and z on the test set. Classification based on y is given by the prediction scores from the FAE classification branch. The inferred z requires an additional classification model and we train a linear SVM using z from the training set and test it on the test set. Predictions based on y and z should thus ideally give perfect and random chance accuracies respectively. Table 3 shows that with SDL, the style feature z's classification performance is close to random guess (10%), and better (closer to random) than that of XCov and DeCov, whilst using with the vanilla FAE with no decorrelation loss, it still contains extensive class information. Meanwhile, the disentangled y provides the highest classification accuracy using our FAE+SDL. The results suggest that our model is more effective than the alternative XCov and DeCov in disentangling latent factors. This is because our SDL does a stochastic approximation of the full-batch statistics, whilst both XCov and DeCov only use information from each mini-batch.

Qualitative results With the style factor disentangled from the class factor, we can use the FAE to transfer styles to a new digit. Given an input image containing a certain digit with certain handwriting style, we can keep the inferred z and change the value y manually to a different digit class. After feeding both the original z and the modified y to the decoder, we can synthesise a new digit with the same style as the input image. Qualitative results are shown in Fig. 6. We see the better disentanglement efficacy of our model in terms of clearer digit reconstruction with clearer style transfer.

5.3. CNN Classifier with SDL

Experiments on object recognition We use CIFAR10 [17] which consists of 60,000 32×32 colour images in 10 categories, with 6000 images per category. We follow the standard experimental setting in [17]. The DNN baseline model used is a 20-layer ResNet [11]. We compares SDL with existing decorrelation loss DeCov [5] and the baseline (with BN but without any decorrelation loss) in Table 4. The proposed SDL leads to a 1.32% performance improvement over the baseline model and also outperforms the alternative DeCov loss by 0.82%.

Person re-identification In this experiment, a CNN classifier is applied to solve a more challenging recognition problem. The person re-identification (Re-ID) problem aims to match pedestrians captured by non-overlapping CCTV cameras[2]. We use one of the biggest and most popu-

[2]Note that although Re-ID can be interpreted as a multi-view learning problem, state-of-the-art approaches treat it as an identity-supervised single-view identity classification problem. [39]; we thus follow this

Method	S-Query		M-Query	
	mAP	R1	mAP	R1
LDEHL [31]	–	59.47	–	–
Siamese LSTM [33]	–	–	35.3	61.6
Gated S-CNN [32]	39.55	65.88	48.45	76.04
CNN Embedding [45]	59.87	79.51	70.33	85.84
Spindle [42]	-	76.9	-	-
HP-net [22]	-	76.9	-	-
OIM [40]	-	82.1	-	-
Re-rank [47]	63.6	77.1	-	-
DPA [43]	63.4	81.0	-	-
SVDNet [29]	62.1	82.3	-	-
ACRN [26]	62.6	83.6	-	-
Context [20]	57.5	80.3	66.7	86.8
JLML [21]	64.4	83.9	74.5	89.7
LSRO [46]	66.1	84.0	**76.1**	88.4
DGDNet*	64.55	85.06	73.30	89.40
DGDNet+DeCov [5]	65.74	85.86	74.72	90.53
DGDNet+SDL	**67.67**	**86.75**	75.77	**91.06**

Table 5: Market-1501 Results. S-Query means Single Query, and M-Query means Multiple Query. '–' indicates no reported result. DGDNet* refers to the basic network used in DGD [39], but trained from scratch only on Market-1501, without multi-task learning through the Domain Guided Dropout layer using six auxiliary datasets for fair comparison with the state-of-the-art.

	CIFAR 10	Market-1501
DeCov [5]	91.62	85.86
DeCovGC	91.86	86.28
DeCovL1	91.90	86.01
SDL	**92.44**	**86.75**

Table 6: Ablation study on the advantage of SDL over DeCov. The CIFAR10 classification results are in classification accuracy (%) and the Market-1501 results are in R1 accuracy (%) under the single query setting.

lar Re-ID benchmarks. **Market-1501** [44] is collected from 6 different cameras. It has 32,668 bounding boxes of 1,501 identities obtained using a Deformable Part Model (DPM) person detector. Following the standards split [44], we use 751 identities with 12,936 images for training and the rest 750 identities with 19,732 images for testing. Experiments are conducted under both the single-query and multi-query evaluation settings. The Rank-1 accuracy is computed to evaluate all the methods. We also calculate the mean average precision (mAP) [44]. For the base model, we use one of the state-of-the-art deep Re-ID models, DGDNet [39], which is built on Inception modules [30]. Our model (DGDNet+SDL) adds SDL on the output of each BN layer in DGDNet during training.

The results are shown in Table 5, along with some recent high performing state-of-the-art alternatives. We can see that: (1) Our model (DGDNet+SDL) outperforms a number of state-of-the-art alternatives. (2) Compared to the base model (DGDNet without decorrelation loss), adding our SDL boosts the performance by a clear margin. (3) When the alternative DeCov loss is added to the base model, its performance is also improved, but by a smaller margin. This result thus indicates that the proposed SDL is more effective than DeCov.

Ablation study Note that SDL differs from DeCov in two aspects: (i) SDL approximates the global covariance by accumulating mini-batch covariance statistics; and (ii) SDL exploits an L_1 instead of L_2 formulation as in DeCov for robustness and correlation sparsity. In order to gain some insight on what contribute to SDL's superior performance, we consider two variants of DeCov [5], called DeCovGC and DeCovL1. DeCovGC is DeCov with added accumulating covariance statistic only while DeCovL1 adopts a $L1$ formulation as in SDL. As shown in Table 6, both DeCov variants have better results than DeCov [5] while SDL (with both accumulating covariance statistic and $L1$ loss) achieves the highest performance among them. It suggests that both differences contribute to the effectiveness of SDL.

6. Conclusions

We have proposed a novel deep CCA model, termed Soft CCA, which provides an efficient and effective solution to deep CCA optimisation by introducing a soft decorrelation loss. Extensive experiments show that the proposed Soft CCA is more effective and scalable than existing CCA variants. Compared to exact whitening solutions, Soft CCA is easy to implement in contemporary learning frameworks, and therefore is promising for enabling practical use of CCA techniques in the deep learning community. Moreover, we demonstrated that as a by-product, the developed SDL loss can be applied beyond CCA as a general purpose decorrelation loss – to any deep learning task where feature decorrelation is required. As case studies, SDL was shown to outperform alternative decorrelation losses in FAE latent factor disentanglement and CNN object and instance recognition.

References

[1] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *ICML*, 2013. 1, 2, 3, 4, 6

[2] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran. Correlational neural networks. *Neural computation*, 2016. 1, 5, 6

single-view approach.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *Intelligent Systems and Technology*, 2011. 5

[4] B. Cheung, J. A. Livezey, A. K. Bansal, and B. A. Olshausen. Discovering hidden factors of variation in deep networks. *ICLR workshop*, 2015. 2, 6, 7

[5] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv*, 2015. 2, 7, 8

[6] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 2

[7] G. H. Golub and H. Zha. The canonical correlations of matrix pairs and their numerical computation. In *Linear algebra for signal processing*. 1995. 1, 3

[8] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014. 1, 2

[9] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. The cmu multi-pose, illumination, and expression (multi-pie) face database. *Technical report, Carnegie Mellon University Robotics Institute. TR-07–08*, 2007. 5

[10] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 2004. 1, 2, 6

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[12] H. Hotelling. Relations between two sets of variates. *Biometrika*, 1936. 1, 2, 6

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *JMLR*, 2015. 2, 3

[14] M. Kan, S. Shan, and X. Chen. Multi-view deep network for cross-view classification. *CVPR*, 2016. 1

[15] T. Karaletsos, S. Belongie, and G. Rätsch. Bayesian representation learning with oracle constraints. *arXiv*, 2015. 2

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2014. 2

[17] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 2, 7

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *IEEE*, 1998. 5, 6

[20] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *CVPR*, 2017. 8

[21] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. *IJCAI*, 2017. 8

[22] X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. *ICCV*, 2017. 8

[23] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv*, 2015. 2

[24] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *NIPS*, 2016. 2

[25] J. Nocedal and S. J. Wright. *Sequential quadratic programming*. Springer, 2006. 2

[26] A. Schumann and R. Stiefelhagen. Person re-identification by deep learning attribute-complementary information. In *CVPR Workshops*, 2017. 8

[27] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014. 2

[28] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, pages 1988–1996. 2014. 2

[29] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval. *ICCV*, 2017. 8

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 8

[31] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, 2016. 8

[32] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, 2016. 8

[33] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang. A siamese long short-term memory architecture for human re-identification. In *ECCV*, 2016. 8

[34] A. Veit, S. Belongie, and T. Karaletsos. Conditional similarity networks. *arXiv*, 2016. 2

[35] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *ICML*, 2015. 6

[36] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *ICASSP*, 2015. 1, 2, 3

[37] W. Wang, R. Arora, K. Livescu, and N. Srebro. Stochastic optimization for deep cca via nonlinear orthogonal iterations. In *Allerton*, 2015. 1, 2, 3, 4, 6

[38] Y. Wen, Z. Li, and Y. Qiao. Latent factor guided convolutional neural networks for age-invariant face recognition. In *CVPR*, 2016. 2

[39] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016. 2, 7, 8

[40] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. Joint detection and identification feature learning for person search. In *CVPR*, 2017. 8

[41] F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *CVPR*, pages 3441–3450, 2015. 1

[42] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017. 8

[43] L. Zhao, X. Li, J. Wang, and Y. Zhuang. Deeply-learned part-aligned representations for person re-identification. *ICCV*, 2017. 8

[44] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 8

[45] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned cnn embedding for person re-identification. *arXiv*, 2016. 8

[46] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, 2017. 8

[47] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. *CVPR*, 2017. 8

[48] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 2

[49] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *NIPS*. 2014. 1, 2